



ATID Application Development Framework Reference Manual–System Service

Revision: Ver. 0.5

Date: September, 2013

ATID Co.,Ltd

Table of Contents

Table of Contents	2
Acronym	6
1 .NET API Reference	8
1.1 Enumerations	8
1.1.1 SYSSVC_RESULT	8
1.1.2 AUDIOCODECCTRL	8
1.1.3 AUDIOCODECVOLUME	8
1.1.4 IPM_PRODUCT_OP	9
1.1.5 KBD_STATUS	9
1.1.6 HW_TYPE_BARCODE	9
1.1.7 HW_TYPE_BLUETOOTH	10
1.1.8 HW_TYPE_CAM	10
1.1.9 HW_TYPE_FINGERPRINT	10
1.1.10 HW_TYPE_GPS	10
1.1.11 HW_TYPE_HF	11
1.1.12 HW_TYPE_MODEM	11
1.1.13 HW_TYPE_MSR	12
1.1.14 HW_TYPE_PRINTER	12
1.1.15 HW_TYPE_SMARTCARD	12
1.1.16 HW_TYPE_UHF	12
1.1.17 HW_TYPE_WLAN	13
1.2 Structures	14
1.2.1 AUDIOCODECVOLCTL	14
1.2.2 VERSION_INFOS	14
1.2.3 HW_OPTIONS	15
1.3 Constants	17
1.3.1 MODE_AC	17
1.3.2 MODE_BAT	17
1.3.3 MODE_KEYLAMP_OFFTIME_START	17
1.3.4 MODE_KEYLAMP_OFFTIME_END	17
1.4 Delegates	18
1.4.1 SleepWakeupNotifyCALLBACK	18
1.4.2 KeyEventNotifyCALLBACK	18
1.5 Methods	19
1.5.1 AudioCodecControl	19
1.5.2 BacklightGetBrightness	19
1.5.3 BacklightReadBrightness	19

1.5.4	BacklightReadTimeoutValue	20
1.5.5	BacklightSetBrightness	20
1.5.6	BacklightWriteBrightness.....	21
1.5.7	BacklightWriteTimeoutValue	21
1.5.8	EnterSuspendState.....	21
1.5.9	SoftReset	22
1.5.10	GetCpuClock.....	22
1.5.11	GetKBDStatus	22
1.5.12	GetModemScreenOffTime	23
1.5.13	GetSoftwareVersion.....	23
1.5.14	GetWlanPowerStatus	23
1.5.15	KeyLampReadOffTimeEnable.....	24
1.5.16	KeyLampReadOffTimeValue	24
1.5.17	KeyLampReadTimeoutValue	25
1.5.18	KeyLampTurnOff	25
1.5.19	KeyLampTurnOn	25
1.5.20	KeyLampWriteOffTimeEnable	26
1.5.21	KeyLampWriteOffTimeValue.....	26
1.5.22	KeyLampWriteTimeoutValue	27
1.5.23	ReadAudioCodecGainLevel.....	27
1.5.24	SetAudioCodecGainLevel	28
1.5.25	SetCpuClock	28
1.5.26	SetKBDStatus.....	28
1.5.27	SetModemScreenOffTime	29
1.5.28	SetVibratorEnable	29
1.5.29	SetWlanPowerEnable.....	30
1.5.30	SleepWakeupNotificationSet	30
1.5.31	SleepWakeupNotificationReset.....	30
1.5.32	SuspendReadTimeoutValue	31
1.5.33	SuspendWriteTimeoutValue	31
1.5.34	WriteAudioCodecGainLevel.....	32
1.5.35	EnableFlashLight.....	32
1.5.36	KeyEventNotificationSet.....	32
1.5.37	KeyEventNotificationReset	33
1.5.38	GetWirelessDevicesPowerState	33
1.5.39	SetWirelessDevicesPowerState.....	33
1.5.40	SetFW_EN	34
1.5.41	SetPWR_EN.....	34
1.5.42	SetRF_5V	35

1.5.43	GetHardwareOptions	35
1.6	Properties	37
1.6.1	DeviceID	37
1.6.2	UUID	37
2	C/C++ API Reference	38
2.1	Enumerations	38
2.1.1	SYSSVC_RESULT	38
2.1.2	AUDIOCODECCTRL	38
2.1.3	AUDIOCODECVOLUME	38
2.1.4	IPM_PRODUCT_OP	39
2.1.5	KBD_STATUS	39
2.2	Structures	40
2.2.1	AUDIOCODECVOLCTL	40
2.2.2	SLEEPWAKEUP_PARAMS	40
2.2.3	HW_OPTIONS	41
2.3	Constants	43
2.3.1	MODE_AC	43
2.3.2	MODE_BAT	43
2.3.3	MODE_KEYLAMP_OFFTIME_START	43
2.3.4	MODE_KEYLAMP_OFFTIME_END	43
2.3.5	WM_SLEEP_WAKEUP_NOTIFY	43
2.4	Callback function definition	43
2.4.1	KEYEVENTCALLBACK	43
2.5	Methods	44
2.5.1	AllocSysSvc	44
2.5.2	DeallocSysSvc	44
2.5.3	AudioCodecControl	44
2.5.4	BacklightGetBrightness	45
2.5.5	BacklightReadBrightness	45
2.5.6	BacklightReadTimeoutValue	45
2.5.7	BacklightSetBrightness	46
2.5.8	BacklightWriteBrightness	46
2.5.9	BacklightWriteTimeoutValue	47
2.5.10	EnterSuspendState	47
2.5.11	SoftReset	47
2.5.12	GetCpuClock	48
2.5.13	GetKBDStatus	48
2.5.14	GetModemScreenOffTime	48
2.5.15	GetSoftwareVersion	49

2.5.16	GetWlanPowerStatus	49
2.5.17	KeyLampReadOffTimeEnable	50
2.5.18	KeyLampReadOffTimeValue	50
2.5.19	KeyLampReadTimeoutValue	51
2.5.20	KeyLampTurnOff	51
2.5.21	KeyLampTurnOn	51
2.5.22	KeyLampWriteOffTimeEnable	52
2.5.23	KeyLampWriteOffTimeValue	52
2.5.24	KeyLampWriteTimeoutValue	53
2.5.25	ReadAudioCodecGainLevel	53
2.5.26	SetAudioCodecGainLevel	53
2.5.27	SetCpuClock	54
2.5.28	SetKBDStatus	54
2.5.29	SetModemScreenOffTime	55
2.5.30	SetVibratorEnable	55
2.5.31	SetWlanPowerEnable	55
2.5.32	SleepWakeupNotificationSet	56
2.5.33	SleepWakeupNotificationReset	56
2.5.34	SuspendReadTimeoutValue	57
2.5.35	SuspendWriteTimeoutValue	57
2.5.36	WriteAudioCodecGainLevel	58
2.5.37	EnableFlashLight	58
2.5.38	KeyEventNotificationSet	58
2.5.39	KeyEventNotificationReset	59
2.5.40	GetWirelessDevicesPowerState	59
2.5.41	SetWirelessDevicesPowerState	60
2.5.42	SetPWR_EN	60
2.5.43	SetFW_EN	61
2.5.44	SetRF_5V	61
2.5.45	GetHardwareOptions.	62

Acronym

modules	descriptions
AADF	ATIDApplication Development Framework

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/25	Draft		Y. J. CHO
0.2	2013/01/25	Update	<ul style="list-style-type: none"> - SoftReset added. - EnableFlashLight added. - KeyEventNotificationSet/Reset added. 	Y. J. CHO
0.3	2013/02/26	Update	- Power Control Function of Wireless Device added.	Y. J. CHO
0.4	2013/05/28	Update	<ul style="list-style-type: none"> -RFID port pin control function added. - GetHardwareOptions function added. 	Y. J. CHO
0.5	2013/09/13	Update	- typo correction.	Y. J. CHO

1 .NET API Reference

1.1 Enumerations

1.1.1 SYSSVC_RESULT

The result of a call to functions.

- **SYSSVC_RESULT_SUCCESS**
Function executed successfully
- **SYSSVC_RESULT_INVALID_ARGS**
Invalid parameter
- **SYSSVC_RESULT_OUTOFMEMORY**
Assign memory failed
- **SYSSVC_RESULT_UNSUPPORTED**
Cannot support function currently
- **SYSSVC_RESULT_FAILURE**
Function executed fail
- **SYSSVC_RESULT_INVALID_DEVICE**
Didn't install WLAN module

1.1.2 AUDIOCODECCTRL

a range of Audio codec value.

- **AUDIOCODECCTRL_READ**
Read Audio codec value
- **AUDIOCODECCTRL_WRITE**
Write Audio codec value to registry
- **AUDIOCODECCTRL_SET**
Apply Audio codec value without save registry

1.1.3 AUDIOCODECVOLUME

type of Audio codec

- **CODEC_MIC_VOLUME**
MIC volume
- **CODEC_SPEAKER_VOLUME**

Speaker volume

1.1.4 IPM_PRODUCT_OP

CPU clock.

- **IPM_OP_104MHZ**

104Mhz

- **IPM_OP_208MHZ**

208Mhz.

- **IPM_OP_416MHZ**

416Msz

- **IPM_OP_624MHZ**

624Mhz

- **IPM_OP_806MHZ**

806Mhz

- **IPM_OP_AUTO**

Setting Clock automatically depending on CPU Usage.

1.1.5 KBD_STATUS

Status value of Key pad status icon.

- **KBD_STATUS_LOWER_CASE**

Lower case

- **KBD_STATUS_NUMERIC**

numeric

- **KBD_STATUS_UPPER_CASE**

Upper case

1.1.6 HW_TYPE_BARCODE

The type of barcode scanner equipped in PDA.

- **_1D**

Motorola 1D Barcode Scanner

- **_2D_HWD**

2D Barcode Scanner with hardware decoder

- **_2D_SW_D_H**

HHP 2D Barcode Scanner with software decoder

- **_2D_SW_D_M**

Motorola 2D Barcode Scanner with software decoder

- **None**

Barcode scanner is not equipped.

1.1.7 HW_TYPE_BLUETOOTH

The type of Bluetooth equipped in PDA.

- **HBM2X1M**

HBM2X1M module is equipped. Use ATID API

- **MD6RF**

MD6RF module is equipped. Use MS Windows API

- **NONE**

Bluetooth is not equipped.

1.1.8 HW_TYPE_CAM

The type of Camera equipped in PDA.

- **OV3640**

Camera is equipped.

- **NONE**

Camera is not equipped.

1.1.9 HW_TYPE_FINGERPRINT

The type of fingerprint identification module equipped in PDA.

- **TFM_100AI**

fingerprint identification module is equipped.

- **NONE**

fingerprint identification module is not equipped.

1.1.10 HW_TYPE_GPS

The type of GPS module equipped in PDA.

- **SIRF3**

SIRF3 module is equipped.

- **UBLUX**

UBLUX module is equipped.

- **NONE**

GPS module is not equipped.

1.1.11 HW_TYPE_HF

The type of HF module equipped in PDA.

- **ESB**

ESB module is equipped.

- **NONE**

HF module is not equipped.

1.1.12 HW_TYPE_MODEM

The type of modem equipped in PDA.

- **AM200**

CDMA 1x modem(Korea-only)

- **AME200**

EVDO modem(Korea-only)

- **HC25**

3G modem

- **HC28**

3G modem(Japan-only)

- **HUAWEI**

EVDO modem(China-only)

- **MC55I**

GSM modem

- **PH8**

3G modem

- **UC864**

3G modem

- **NONE**

Modem is not equipped.

1.1.13 HW_TYPE_MSR

The type of Magnetic Stripe Reader module equipped in PDA.

- **MCR_3006**

MCR_3006 module is equipped.

- **NONE**

MSR is not equipped.

1.1.14 HW_TYPE_PRINTER

The type of Printer module equipped in PDA.

- **LTPD245**

LTPD245 module is equipped.

- **NONE**

Printer module is not equipped.

1.1.15 HW_TYPE_SMARTCARD

The type of Smart Card module equipped in PDA.

- **_73S12XXF**

73S12XXF module is equipped.

- **EICM1000**

EICM1000 module is equipped.

- **NONE**

Smart Card module is not equipped.

1.1.16 HW_TYPE_UHF

The type of UHF module equipped in PDA.

- **H1000**

H1000 module is equipped.

- **I2000**

I2000 module is equipped.

- **I900**

I900 module is equipped.

- **NONE**

UHF module is not equipped.

1.1.17 HW_TYPE_WLAN

The type of WLAN module equipped in PDA.

- **MCF10G**

MCF10G module is equipped.

- **MSD10G**

MSD10G module is equipped.

- **MSD30AG**

MSD30AG module is equipped.

- **NONE**

WLAN module is not equipped..

1.2 Structures

1.2.1 AUDIOCODECVOLCTL

Audio codec structure to adjust volume (not used)

Public struct `AUDIOCODECVOLCTL`

```
{
    AUDIOCODECCTRL AudioCodecCtrl
    AUDIOCODECVOLUME AudioCodecVolume;
    IntPtr pVolumeLevel;
};
```

- **AudioCodecCtrl**
A range of application of value.
- **AudioCodecVolume**
Type of codec
- **pVolumeLevel**
value

1.2.2 VERSION_INFOS

S/W version structure (not used)

Public struct `VERSION_INFOS`

```
{
    byte[] ATBTVer;9
    byte[] BSPVer;
    byte[] CEOSVer;
    byte[] SDKVer;
};
```

- **ATBTVer**
Boot Version
- **BSPVer**
BSP Version
- **CEOSVer**
CE OS Version
- **SDKVer**
SDK Version

1.2.3 HW_OPTIONS

Structure used to check Hardware option

Using GetHardwareOptions function

Public struct [HW_OPTIONS](#)

```
{  
    HW\_TYPE\_BARCODE Barcode;  
    HW\_TYPE\_BLUETOOTH Bluetooth;  
    HW\_TYPE\_CAM Camera;  
    HW\_TYPE\_FINGERPRINT FingerPrint;  
    HW\_TYPE\_GPS GPS;  
    HW\_TYPE\_HF HF;  
    HW\_TYPE\_Modem Modem;  
    HW\_TYPE\_MSR MSR;  
    HW\_TYPE\_PRINTER Printer;  
    HW\_TYPE\_SMARTCARD SmartCard;  
    HW\_TYPE\_UHF UHF;  
    HW\_TYPE\_WLAN WLAN;  
};
```

- **Barcode**

The type of Barcode Scanner equipped now.

- **Bluetooth**

The type of Bluetooth equipped now.

- **Camera**

The type of Camera equipped now

- **FingerPrint**

The type of FingerPrint equipped now

- **GPS**

The type of GPS equipped now

- **HF**

The type of HF equipped now

- **Modem**

The type of Modem equipped now

- **MSR**

The type of MSR equipped now

- **Printer**

The type of Printer equipped now

- **SmartCard**

The type of SmartCard equipped now

- **UHF**

The type of UHF equipped now

- **WLAN**

The type of WLAN equipped now

1.3 Constants

1.3.1 MODE_AC

Using AC

1.3.2 MODE_BAT

Using battery.

1.3.3 MODE_KEYLAMP_OFFTIME_START

Start of key lamp off (not used)

1.3.4 MODE_KEYLAMP_OFFTIME_END

End of key lamp off (not used)

1.4 Delegates

1.4.1 SleepWakeupNotifyCALLBACK

the callback delegate called when PDA entering Suspend State (sleep) or waking up from Suspend State (wake up). When PDA is changed to sleep or Wakeup status, if you need to do a specific task, you must set the callback delegate using SleepWakeupNotificationSet () function. if value of bEnterSleep is true, PDA is entering sleep status, if false, the wakeup state.

Public delegate void **SleepWakeupNotifyCALLBACK**(bool bEnterSleep);

1.4.2 KeyEventNotifyCALLBACK

Key Pad의 Up/Down is informed not using Application Window of OnKeyDown, OnKeyUp but registering delegate Function.

KeyValue : reference to "ATID KeyCode_Rev_0_3.docx".

KeyState : 0x0101 Up
0x0100 Down

Public delegate void **KeyEventNotifyCALLBACK**(int KeyValue, int KeyState);

1.5 Methods

1.5.1 AudioCodecControl

Set the Audio codec volume (not used)

```
SYSSVC_RESULT AudioCodecControl(
    refAUDIOCODECVOLCTL AudioCodecVolCtrl
);
```

Parameters

AudioCodecVolCtrl

Control value of Codec Volume.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.2 BacklightGetBrightness

Get the value of brightness in the current power mode.

```
SYSSVC_RESULT BacklightGetBrightness(
    refushort nBrightness
);
```

Parameters

nBrightness

backlight value

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.3 BacklightReadBrightness

Get the value of brightness that set at relevant power mode.

```
SYSSVC_RESULT BacklightReadBrightness(
    byte cMode
    refushort nBrightness
);
```

Parameters

cMode

the power mode, that would like to get its value of brightness, will be transferred.

nBrightness

the power mode, that would like to get its value of brightness, will be transferred.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.4 BacklightReadTimeoutValue

Get the values of currently set time-out for each power mode.

```
SYSSVC_RESULT BacklightReadTimeoutValue(
    byte cMode
    refushort nSeconds
);
```

Parameters

cMode

transfer the power mode that will get the value of Time-out.

nBrightness

transfer the form of UInt16 parameter to ref form that which will be stored the value of time-out that according to the each Power *Mode*.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if the user's input doesn't exist during the nSeconds value, the backlight illumination lights turned off, and If nSeconds is 0, the backlight does not turn off.

1.5.5 BacklightSetBrightness

Set the value of brightness in the current power mode.

```
SYSSVC_RESULT BacklightSetBrightness(
    ushort nBrightness
);
```

Parameters

nBrightness

could set the backlight brightness values between 0(Min.) to 100(Max.).

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.6 BacklightWriteBrightness

Get the value of brightness for relevant power mode.

```
SYSSVC_RESULT BacklightWriteBrightness(  
    byte cMode  
    ushort nBrightness  
);
```

Parameters

cMode

transfer the power mode which will be set the value of brightness.

nBrightness

transfer the brightness values between 27 ~ 100 which will be set into power mode that transferred to the *Mode*.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.7 BacklightWriteTimeoutValue

Reading Timeout Value of Backlight about Mode(AC/Battery)

```
SYSSVC_RESULT BacklightWriteTimeoutValue(  
    byte cMode  
    ushort nSeconds  
);
```

Parameters

cMode

Mode(AC/Battery) value.

nBrightness

Time-out value of Backlight(seconds)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if the user's input doesn't exist during the nSeconds value, the backlight illumination lights turned off, and If nSeconds is 0, the backlight does not turn off.

1.5.8 EnterSuspendState

Entering the Suspend state(sleep) mode.

```
SYSSVC_RESULT EnterSuspendState();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.9 SoftReset

Soft(warm) resetting.

SYSSVC_RESULT EnterSuspendState();

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

1.5.10 GetCpuClock

Get the currently setting mode of CPU clock.

SYSSVC_RESULT GetCpuClock(
 refIPM_PRODUCT_OP Clock
);

Parameters

Clock

Cpu clock

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE

1.5.11 GetKBDSStatus

Getting the input mode value of Hardware Keypad

SYSSVC_RESULT GetKBDSStatus(
 refKBD_STATUS pKbdStatus
);

Parameters

pKbdStatus

the input mode value of Hardware Keypad

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Reading the value of changing mode when inputting button of Keypad

1.5.12 GetModemScreenOffTime

Get the currently set screen off time.

```
SYSSVC_RESULT GetModemScreenOffTime(
    refushort nSeconds
);
```

Parameters

nSeconds
screen off time(seconds)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE

1.5.13 GetSoftwareVersion

Get the S/W versions (not used)

```
SYSSVC_RESULT GetSoftwareVersion(
    refVERSION_INFOS VersionInfos
);
```

Parameters

VersionInfos
S/W Version

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.14 GetWlanPowerStatus

Get the current state of the WLAN module power.

```
SYSSVC_RESULT GetWlanPowerStatus(
    refbool bEnable
);
```

Parameters

bEnable
power status fo WLAN Device.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if executed successfully.

Notes

TRUE: WLAN module applied voltage.

FALSE: WLAN module did not applied voltage.

1.5.15 KeyLampReadOffTimeEnable

Read the currently set value of key lamp off function for conserving battery power

```
SYSSVC_RESULT KeyLampReadOffTimeEnable(
    refbool bEnable
);
```

Parameters

bEnable

whether to use Key Lamp Off function

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If function of key Lamp Off is ON, bEnable value is true, if OFF, false.

1.5.16 KeyLampReadOffTimeValue

Read the set value of starting and ending time that period of not use key lamp.

```
SYSSVC_RESULT KeyLampReadOffTimeValue(
    refushort nHoursFrom
    refushort nMinutesFrom
    refushort nHoursTo
    refushort nMinutesTo
);
```

Parameters

nHoursFrom

the value of the starting time (hour) that period of not using the key lamp

nMinutesFrom

the value of the starting time (Minutes) that period of not using the key lamp

nHoursTo

the value of the ending time (hour) that period of not using the key lamp

nMinutesTo

the value of the ending time (minutes) that period of not using the key lamp

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.17 KeyLampReadTimeoutValue

Read the time that set to off the key lamp after users pressed final button.

```
SYSSVC_RESULT KeyLampReadTimeoutValue(  
    refushort nSeconds  
);
```

Parameters

nSeconds
time(in seconds) until Key Lamp is OFF

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.18 KeyLampTurnOff

Switch key pad lamp off.

```
SYSSVC_RESULT KeyLampTurnOff();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.19 KeyLampTurnOn

Switch Key pad lamp on.

```
SYSSVC_RESULT KeyLampTurnOn();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.20 KeyLampWriteOffTimeEnable

Enable or disable the key off function in the period of setting time to conserve battery power.

```
SYSSVC_RESULT KeyLampWriteOffTimeEnable(
    bool bEnable
);
```

Parameters

bEnable

TRUE : enable the key off function, FALSE: disable the key off function.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If bEnable value is true, Key Lamp Off function is ON, if false, OFF.

Only in support of Windows CE.

1.5.21 KeyLampWriteOffTimeValue

Set does not use the Key lamp time period. Key lamp will not work when press the button during the current set timing.

```
SYSSVC_RESULT KeyLampWriteOffTimeValue(
    ushort nHoursFrom
    ushort nMinutesFrom
    ushort nHoursTo
    ushort nMinutesTo
);
```

Parameters

nHoursFrom

parameter that transfer the value of the starting time (hour) of not use key lamp.

nMinutesFrom

parameter that transfer the value of the starting time (Minutes) of not use key lamp

nHoursTo

parameter that transfer the value of the ending time (hour) of not use key lamp.

nMinutesTo

parameter that transfer the value of the ending time (Minutes) of not use key lamp.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.22 KeyLampWriteTimeoutValue

Set the time to off the lamp, when after pressed the button.

```
SYSSVC_RESULT KeyLampWriteTimeoutValue(
    ushort nSeconds
);
```

Parameters

nSeconds

after pressed the button, transfer the time by second to off the key lamp automatically.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.23 ReadAudioCodecGainLevel

Get the codec gain level.

```
SYSSVC_RESULT ReadAudioCodecGainLevel(
    AUDIOCODECVOLUME Type,
    refushort GainLevel
);
```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.24 SetAudioCodecGainLevel

Set the codec gain level

```
SYSSVC_RESULT SetAudioCodecGainLevel(
    AUDIOCODECVOLUME Type,
    ushort GainLevel
);
```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.25 SetCpuClock

Set the CPU clock

```
SYSSVC_RESULT SetCpuClock(
    IPM_PRODUCT_OP Clock
);
```

Parameters

Clock

Cpu clock

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.26 SetKBDSStatus

Set the status of KBD to uppercase, lowercase, and number.

```
SYSSVC_RESULT SetKBDSStatus(
    KBD_STATUS KbdStatus
);
```

Parameters

pKbdStatus

the inputting mode of Hardware KeyPad.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Setting changing mode(number/lowercase/uppercase) when inputting
"1 <-> ALP" button of Keypad

1.5.27 SetModemScreenOffTime

Set the off the screen timing (seconds) when after entered the communicate mode.

```
SYSSVC_RESULT SetModemScreenOffTime(
    ushort nSeconds
);
```

Parameters

nSeconds

after enter the call mode and setting *nSeconds* , the screen will be off.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.28 SetVibratorEnable

Enable or disable the Vibrator motor.

```
SYSSVC_RESULT SetVibratorEnable(
    bool bEnable
);
```

Parameters

bEnable

TRUE: enable to use motor, FALSE: disable to use the motor.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

bEnable value True : ON

bEnable value False : OFF

1.5.29 SetWlanPowerEnable

Apply or remove the power of the WLAN module.

```
SYSSVC_RESULT SetWlanPowerEnable(
    bool bEnable
);
```

Parameters

bEnable

the flag parameter which used to apply or remove the voltage of the WLAN module.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if executed successfully,

Notes

bEnable value True : WLAN ON

bEnable value False : WLAN OFF

1.5.30 SleepWakeupNotificationSet

Register the application Callback function which will receive the Sleep/wakeup notification service.

```
SYSSVC_RESULT SleepWakeupNotificationSet(
    SleepWakeupNotifyCALLBACK CallbackProc
);
```

Parameters

CallbackProc

callback delegate which will be executed in sleep/wakeup mode(reference to 1.4.1 SleepWakeupNotifyCALLBACK)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

When PDA becomes Sleep or Wakeup, if there is contents to be processed in application, you can process to register callback delegate using this function.

Only in support of Windows CE

1.5.31 SleepWakeupNotificationReset

Remove the sleep/wakeup notification service's Callback which has registered.

```
SYSSVC_RESULT SleepWakeupNotificationReset();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if you register for the service by using the function of SleepWakeupNotificationSet registration should be unregistered using this function.

1.5.32 SuspendReadTimeoutValue

Reading time(in seconds) from the last input to entering suspend state(Sleep)

```
SYSSVC_RESULT SuspendReadTimeoutValue(
    byte cMode,
    refushort nSeconds
);
```

Parameters

cMode

mode(ac/Battery) value.

nSeconds

entering standby time(in seconds) to suspend state.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If nSeconds is 0, not enter suspend state.

1.5.33 SuspendWriteTimeoutValue

setting time(in seconds) from the last input to entering suspend state(Sleep).

```
SYSSVC_RESULT SuspendWriteTimeoutValue(
    byte cMode,
    ushort nSeconds
);
```

Parameters

cMode

Mode(AC/Battery) value.

nSeconds

entering standby time(in seconds) to suspend state.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If nSeconds is 0, not enter suspend state.

1.5.34 WriteAudioCodecGainLevel

Set the codecgain level, and save to registry.

```
SYSSVC_RESULT WriteAudioCodecGainLevel(
    AUDIOCODECVOLUME Type,
    ushort GainLevel
);
```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

1.5.35 EnableFlashLight

Applying on/off to Flashlight

```
SYSSVC_RESULT EnableFlashLight(
    bool Enable,
);
```

Parameters

Enable

Flashlight status

True : trun on, False : turn off

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.36 KeyEventNotificationSet

Registering delegate to get key Event, starting Key Event notification service, regardless of Windows Form

```
SYSSVC_RESULT KeyEventNotificationSet(
    KeyEventNotifyCALLBACK CallbackProc,
```


);

Parameters

CallbackProc

Delegate which will receive Key Code, when Key event occurs.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.37 KeyEventNotificationReset

Stopping service of Key Event notification.

SYSSVC_RESULT KeyEventNotificationReset();

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.38 GetWirelessDevicesPowerState

Reading power status about WiFi, Phone, Bluetooth.

SYSSVC_RESULT GetWirelessDevicesPowerState(
 int Device,
 refint State
);

Parameters

Device

0 : Wifi

1 : Phone

2 : Bluetooth

State

0 : Off

1 : On

2 : Discoverable(Bluetooth)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.39 SetWirelessDevicesPowerState

Turning on/off power about WiFi, Phone, Bluetooth

```
SYSSVC_RESULT SetWirelessDevicesPowerState(
    int Device,
    int State
);
```

Parameters

Device

- 0 : Wifi
- 1 : Phone
- 2 : Bluetooth

State

- 0 : Off
- 1 : On
- 2 : Discoverable(Bluetooth)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

1.5.40 SetFW_EN

Enable/Disable FW_EN pin of RFID port on the back of the body.

```
SYSSVC_RESULT SetFW_EN(
    bool Enable
);
```

Parameters

Enable

- True : Enable
- False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when other than UHF module is attached to RFID port, so Never use when UHF module is equipped.

1.5.41 SetPWR_EN

Enable/Disable PWR_EN pin of RFID port on the back of the body.

```
SYSSVC_RESULT SetPWR_EN(
    bool Enable
```

);

Parameters

Enable

True : Enable

False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when a device aside from UHF module is attached to RFID port, so Never use when UHF module is equipped.

1.5.42 SetRF_5V

Enable/Disable RF_5V pin of RFID port on the back of the body.

```
SYSSVC_RESULT SetRF_5V(
    boolEnable
);
```

Parameters

Enable

True : Enable

False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when a device aside from UHF module is attached to RFID port, so Never use when UHF module is equipped.

In order to communicate with the serial port when a device aside from UHF module is equipped, be sure to enable RF_5V.

1.5.43 GetHardwareOptions

Get the hardware option equipped in PDA.

```
SYSSVC_RESULT GetHardwareOptions(
    refHW_OPTIONS Options
);
```

);

Parameters

Options

Structure where hardware option will be stored.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

1.6 Properties

1.6.1 DeviceID

Get the Device ID

```
string DeviceID{ get; }
```

1.6.2 UUID

Get the UUID

```
string UUID{ get; }
```

2 C/C++ API Reference

2.1 Enumerations

2.1.1 SYSSVC_RESULT

The result of a call to functions.

- **SYSSVC_RESULT_SUCCESS**
Function executed successfully
- **SYSSVC_RESULT_INVALID_ARGS**
Invalid parameter
- **SYSSVC_RESULT_OUTOFMEMORY**
Assign memory failed
- **SYSSVC_RESULT_UNSUPPORTED**
Cannot support function currently
- **SYSSVC_RESULT_FAILURE**
Function executed fail
- **SYSSVC_RESULT_INVALID_DEVICE**
Didn't install WLAN module

2.1.2 AUDIOCODECCTRL

the range of Audio codec value.

- **AUDIOCODECCTRL_READ**
Read Audio codec value
- **AUDIOCODECCTRL_WRITE**
Write Audio codec value to registry
- **AUDIOCODECCTRL_SET**
Apply Audio codec value without save registry

2.1.3 AUDIOCODECVOLUME

type of Audio codec

- **CODEC_MIC_VOLUME**
MIC volume
- **CODEC_SPEAKER_VOLUME**

Speaker volume

2.1.4 IPM_PRODUCT_OP

CPU clock.

- **IPM_OP_104MHZ**

104Mhz

- **IPM_OP_208MHZ**

208Mhz.

- **IPM_OP_416MHZ**

416Msz

- **IPM_OP_624MHZ**

624Mhz

- **IPM_OP_806MHZ**

806Mhz

- **IPM_OP_AUTO**

Setting Clock automatically depending on CPU Usage.

2.1.5 KBD_STATUS

Status value of Key pad status icon.

- **KBD_STATUS_LOWER_CASE**

Lower case

- **KBD_STATUS_NUMERIC**

numeric

- **KBD_STATUS_UPPER_CASE**

Upper case

2.2 Structures

2.2.1 AUDIOCODECVOLCTL

Audio codec structure to adjust volume (not used)

Public struct [AUDIOCODECVOLCTL](#)

```
{
    AUDIOCODECCTRL AudioCodecCtrl
    AUDIOCODECVOLUME AudioCodecVolume;
    IntPtr pVolumeLevel;
};
```

- **AudioCodecCtrl**

A range of application of value.

- **AudioCodecVolume**

Type of codec

- **pVolumeLevel**

value

2.2.2 SLEEPWAKEUP_PARAMS

Structure that will inform sleep/wakeup status of PDA

typedef struct

```
{
    BOOLbEnterSleep;
    BOOLbBatteryDetect;
} SLEEPWAKEUP\_PARAMS;
```

- **bEnterSleep**

value of sleep status

TRUE : Sleep

False : wakeup

- **bBatteryDetect**

value the battery equipped status

2.2.3 HW_OPTIONS

Structure used to check Hardware Option.

Using GetHardwareOptions function.

```
typedef struct
{
    BYTE Barcode;
    BYTE Bluetooth;
    BYTE Camera;
    BYTE FingerPrint;
    BYTE GPS;
    BYTE HF;
    BYTE Modem;
    BYTE MSR;
    BYTE Printer;
    BYTE SmartCard;
    BYTE UHF;
    BYTE WLAN;
}HW_OPTIONS;
```

- Barcode

- 0 :barcode scanner is not equipped.
- 1 :Motorola 1D Barcode Scanner
- 2 :2D Barcode Scanner with hardware decoder
- 3 :HHP 2D Barcode Scanner with software decoder
- 4 :Motorola 2D Barcode Scanner with software decoder

- Bluetooth

- 0 : Bluetooth is not equipped.
- 1 :HBM2X1M module is equipped. Use ATID API
- 2 :MD6RF module is equipped. Use MS Windows API

- Camera

- 0 : Camera is not equipped.
- 1 : OV3640 module is equipped.

- FingerPrint

- 0 :FingerPrint module is not equipped.
- 1 : TFM_100AI module is equipped.

- GPS

- 0 : GPS is not equipped.
- 1 : UBLUX module is equipped.

2 : SIRF3 module is equipped

- **HF**

0 : HF is not equipped.

1 : ESB module is equipped

- **Modem**

0 : Modem is not equipped.

1 : AM200 is equipped.

2 : AME200 is equipped.

3 : UC864 is equipped.

4 : HC25 is equipped.

5 : HC28 is equipped.

6 : PH8 is equipped.

7 : MC55I is equipped.

4 : HUAWEI is equipped.

- **MSR**

0 : MSR is not equipped.

1 : MCR_3006 module is equipped

- **Printer**

0 : printer is not equipped.

1 : LTPD245 is equipped

- **SmartCard**

0 : Smart Card is not equipped.

1 : 73S12XXF is equipped

2 : EICM1000 is equipped

- **UHF**

0x30 : UHF is not equipped.

0x31 : H1000 is equipped

0x32 : I2000 is equipped

0x33 : I900 is equipped

- **WLAN**

0 : WLAN is not equipped.

1 : MCF10G is equipped

2 : MSD10G is equipped

3 : MSD30AG is equipped

2.3 Constants

2.3.1 MODE_AC

Using AC

2.3.2 MODE_BAT

Using battery.

2.3.3 MODE_KEYLAMP_OFFTIME_START

Start of key lamp off (not used)

2.3.4 MODE_KEYLAMP_OFFTIME_END

End of key lamp off (not used)

2.3.5 WM_SLEEP_WAKEUP_NOTIFY

PDA가 Sleep 또는 Wakeup 될때, Application으로통지되는 message.

Message informed to application, when PDA becomes Sleep or Wakeup,

2.4 Callback function definition

2.4.1 KEYEVENTCALLBACK

Pad Up / Down status is informed registering a Callback function, not using WM_KEYUP, WM_KEYDOWN messages which is informed to Application Window

KeyCode : reference to "ATID KeyCode_Rev_0_3.docx".

KeyState : 0x0101 Up

0x0100 Down

- **typedef void (CALLBACK* KEYEVENTCALLBACK)(DWORD keyCode, DWORD keyState);**

2.5 Methods

2.5.1 AllocSysSvc

Allocating resources for system service.

```
SYSSVC_RESULT AllocSysSvc();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

You must call this function first to use System Service API.

2.5.2 DeallocSysSvc

Deallocating resources of system service.

```
SYSSVC_RESULT DeallocSysSvc();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

You must call this function when not using System Service API anymore.

2.5.3 AudioCodecControl

Set the Audio codec volume (not used)

```
SYSSVC_RESULT AudioCodecControl(  
    refAUDIOCODECVOLCTL AudioCodecVolCtrl  
);
```

Parameters

AudioCodecVolCtrl

Control value of Codec Volume.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.4 BacklightGetBrightness

Get the value of brightness in the current power mode.

```
SYSSVC_RESULT BacklightGetBrightness(
    refushort nBrightness
);
```

Parameters

nBrightness
backlight value

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.5 BacklightReadBrightness

Get the value of brightness that set at relevant power mode.

```
SYSSVC_RESULT BacklightReadBrightness(
    charMode
    WORD* npBrightness
);
```

Parameters

Mode
Mode(AC/Battery) value.
npBrightness
backlight brightness value.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.6 BacklightReadTimeoutValue

Reading time(in seconds) from the last input of user to backlight off about Mode(AC/Battery)

```
SYSSVC_RESULT BacklightReadTimeoutValue(
```

```
charMode
WORD*pnSeconds
);
```

Parameters

Mode

Mode(AC/Battery) 값.

pnBrightness

waiting time(in seconds) until backlight turns off

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if the user's input doesn't exist during the pnSeconds value, the backlight illumination lights turns off, and If pnSeconds is 0, the backlight does not turn off.

2.5.7 BacklightSetBrightness

Set the value of brightness in the current power mode.

```
SYSSVC_RESULT BacklightSetBrightness(
    WORD nBrightness
);
```

Parameters

nBrightness

set the backlight brightness values between 0(Min.) to 100(Max.).

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.8 BacklightWriteBrightness.

Set the backlight brightness value and saving it in registry

```
SYSSVC_RESULT BacklightWriteBrightness(
    charMode
    WORD nBrightness
);
```

Parameters

Mode

Mode(AC/Battery) value.

nBrightness

backlight brightness value.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.9 BacklightWriteTimeoutValue

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

```
SYSSVC_RESULT BacklightWriteTimeoutValue(
    charMode
    WORD nSeconds
);
```

Parameters

Mode

Mode(AC/Battery) value.

nSeconds

Waiting time until backlight turns off(in seconds)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if the user's input doesn't exist during the nSeconds value, the backlight illumination lights turns off, and If nSeconds is 0, the backlight does not turn off.

2.5.10 EnterSuspendState

Entering the Suspend state(sleep) mode.

```
SYSSVC_RESULT EnterSuspendState();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.11 SoftReset

Soft(warm) resetting

```
SYSSVC_RESULT EnterSuspendState();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.12 GetCpuClock

Get the currently setting mode of CPU clock.

```
SYSSVC_RESULT GetCpuClock(
    IPM_PRODUCT_OP* pClock
);
```

Parameters

pClock

Cpu clock

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE

2.5.13 GetKBDDStatus

Getting the input mode value of Hardware Keypad

```
SYSSVC_RESULT GetKBDDStatus(
    KBD_STATUS* pkbdStatus
);
```

Parameters

pkbdStatus

the input mode value of Hardware Keypad

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes.

Reading the value of changing mode(numeric/lowercase/uppercase) when inputting "<->ALP" Button of Keypad

2.5.14 GetModemScreenOffTime.

Get the currently set screen off time when the line is busy.

```
SYSSVC_RESULT GetModemScreenOffTime(
    WORD* pnSeconds
```


);

Parameters

pnSeconds

time until backlight power is off(in seconds)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully..

Notes

Only in support of Windows CE

2.5.15 GetSoftwareVersion

Get the S/W versions (not used)

```
SYSSVC_RESULT GetSoftwareVersion(
    refVERSION_INFOS VersionInfos
);
```

Parameters

VersionInfos

S/W Version

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.16 GetWlanPowerStatus

Get the current state of the WLAN module power.

```
SYSSVC_RESULT GetWlanPowerStatus(
    refbool bEnable
);
```

Parameters

bEnable

power status fo WLAN Device.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if executed successfully.

Notes

TRUE: WLAN module applied voltage.

FALSE: WLAN module did not applied voltage.

2.5.17 KeyLampReadOffTimeEnable

Read the currently set value of key lamp off function for conserving battery power

```
SYSSVC_RESULT KeyLampReadOffTimeEnable(
    refbool bEnable
);
```

Parameters

bEnable

whether to use Key Lamp Off function

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If function of key Lamp Off is ON, bEnable value is true, if OFF, false.

Only in support of Windows CE.

2.5.18 KeyLampReadOffTimeValue

Read the set value of starting and ending time that period of not use key lamp.

```
SYSSVC_RESULT KeyLampReadOffTimeValue(
    WORD* pnHoursFrom
    WORD* pnMinutesFrom
    WORD* pnHoursTo
    WORD* pnMinutesTo
);
```

Parameters

pnHoursFrom

the value of the starting time (hour) that period of not using the key lamp

pnMinutesFrom

the value of the starting time (Minutes) that period of not using the key lamp

pnHoursTo

the value of the ending time (hour) that period of not using the key lamp

pnMinutesTo

the value of the ending time (minutes) that period of not using the key lamp

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.19 KeyLampReadTimeoutValue

Read the time that set to off the key lamp after users pressed final button.

```
SYSSVC_RESULT KeyLampReadTimeoutValue(  
    WORD*pnSeconds  
);
```

Parameters

pnSeconds

time(in seconds) until Key Lamp is OFF

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.20 KeyLampTurnOff

Switch key pad lamp off.

```
SYSSVC_RESULT KeyLampTurnOff();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.21 KeyLampTurnOn

Switch Key pad lamp on.

```
SYSSVC_RESULT KeyLampTurnOn();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.22 KeyLampWriteOffTimeEnable

Enable or disable the key off function in the period of setting time to conserve battery power.

```
SYSSVC_RESULT KeyLampWriteOffTimeEnable(
    bool bEnable
);
```

Parameters

bEnable

TRUE : enable the key off function, FALSE: disable the key off function.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If bEnable value is true, Key Lamp Off function is ON, if false, OFF.

Only in support of Windows CE.

2.5.23 KeyLampWriteOffTimeValue

Set does not use the Key lamp time period. Key lamp will not work when press the button during the current set timing.

```
SYSSVC_RESULT KeyLampWriteOffTimeValue(
    WORD nHoursFrom
    WORD nMinutesFrom
    WORD nHoursTo
    WORD nMinutesTo
);
```

Parameters

nHoursFrom

parameter that transfer the value of the starting time (hour) of not use key lamp.

nMinutesFrom

parameter that transfer the value of the starting time (Minutes) of not use key lamp

nHoursTo

parameter that transfer the value of the ending time (hour) of not use key lamp.

nMinutesTo

parameter that transfer the value of the ending time (Minutes) of not use key lamp.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.24 KeyLampWriteTimeoutValue

Set the time to off the lamp, when after pressed the button.

```
SYSSVC_RESULT KeyLampWriteTimeoutValue(  
    WORD nSeconds  
);
```

Parameters

nSeconds

after pressed the button, transfer the time by second to off the key lamp automatically.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.25 ReadAudioCodecGainLevel

Get the codec gain level.

```
SYSSVC_RESULT ReadAudioCodecGainLevel(  
    AUDIOCODECVOLUME Type,  
    USHORT* pGainLevel  
);
```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.26 SetAudioCodecGainLevel

Set the codec gain level

```
SYSSVC_RESULT SetAudioCodecGainLevel(  

```

```

        AUDIOCODECVOLUME Type,
        USHORT GainLevel
    );

```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.27 SetCpuClock

Set the CPU clock

```

SYSSVC_RESULT SetCpuClock(
    IPM_PRODUCT_OP Clock
);

```

Parameters

Clock

Cpu clock

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.28 SetKBDDStatus

Set the status of KBD to uppercase, lowercase, and number.

```

SYSSVC_RESULT SetKBDDStatus(
    KBD_STATUS KbdStatus
);

```

Parameters

pKbdStatus

the inputting mode of Hardware KeyPad.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Setting changing mode(number/lowercase/uppercase) when inputting
"1<->ALP" button of Keypad

2.5.29 SetModemScreenOffTime

Set the off the screen timing (seconds) when after entered the communicate mode.

```
SYSSVC_RESULT SetModemScreenOffTime(  
    WORD nSeconds  
);
```

Parameters

nSeconds

after enter the call mode and setting *nSeconds* , the screen will be off.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.30 SetVibratorEnable

Enable or disable the Vibrator motor.

```
SYSSVC_RESULT SetVibratorEnable(  
    bool bEnable  
);
```

Parameters

bEnable

TRUE: enable to use motor, FALSE: disable to use the motor.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

bEnable value True : ON

bEnable value False : OFF

2.5.31 SetWlanPowerEnable

Apply or remove the power of the WLAN module.

```
SYSSVC_RESULT SetWlanPowerEnable(  
    bool bEnable  
);
```

```
bool bEnable
);
```

Parameters

bEnable

the flag parameter which used to apply or remove the voltage of the WLAN module.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if executed successfully,

Notes

bEnable value True : WLAN ON

bEnable value False : WLAN OFF

2.5.32 SleepWakeupNotificationSet

Register the application Callback function which will receive the Sleep/wakeup notification service.

```
SYSSVC_RESULT SleepWakeupNotificationSet(
    HWND hWnd
);
```

Parameters

hWnd

Window handle of application which will be informed messages in sleep/wakeup mode.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

When PDA becomes Sleep or Wakeup, if there is contents to be processed in application, you can process to register callback delegate using this function.

Only in support of Windows CE

2.5.33 SleepWakeupNotificationReset

Remove the sleep/wakeup notification service's Callback which has registered.

```
SYSSVC_RESULT SleepWakeupNotificationReset(
    HWND hWnd
);
```

Parameters

hWnd

Window handle of application which will remove notification services

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

if you register for the service by using the function of SleepWakeupNotificationSet registration should be unregistered using this function.

Only in support of Windows CE.

2.5.34 SuspendReadTimeoutValue

Reading time(in seconds) from the last input to entering suspend state(Sleep)

```
SYSSVC_RESULT SuspendReadTimeoutValue(
    charMode,
    WORD*pnSeconds
);
```

Parameters

cMode

mode(ac/Battery) value.

nSeconds

entering standby time(in seconds) to suspend state.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If nSeconds is 0, not enter suspend state.

2.5.35 SuspendWriteTimeoutValue

setting time(in seconds) from the last input to entering suspend state(Sleep).

```
SYSSVC_RESULT SuspendWriteTimeoutValue(
    charMode,
    WORD nSeconds
);
```

Parameters

cMode

Mode(AC/Battery) value.

nSeconds

entering standby time(in seconds) to suspend state.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

If nSeconds is 0, not enter suspend state.

2.5.36 WriteAudioCodecGainLevel

Set the codecgain level, and save to registry.

```
SYSSVC_RESULT WriteAudioCodecGainLevel(
    AUDIOCODECVOLUME Type,
    ushort GainLevel
);
```

Parameters

Type

Type of Audio codec.

GainLevel

Gain level.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

Notes

Only in support of Windows CE.

2.5.37 EnableFlashLight

Applying on/off to Flashlight

```
SYSSVC_RESULT EnableFlashLight(
    bool Enable,
);
```

Parameters

Enable

Flashlight status

True : trun on, False : turn off

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.38 KeyEventNotificationSet

Registering delegate to get key Event, starting Key Event notification service, regardless of Windows Form

```
SYSSVC_RESULT KeyEventNotificationSet(
    KEYEVENTCALLBACK pCallback,
    HWND hWnd,
);
```

Parameters

CallbackProc

Callback function which will receive Key Code, when Key event occurs.

hWnd

setting to NULL

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.39 KeyEventNotificationReset

Stopping service of Key Event notification.

```
SYSSVC_RESULT KeyEventNotificationReset();
```

Parameters

None

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.40 GetWirelessDevicesPowerState

Reading power status about WiFi, Phone, Bluetooth.

```
SYSSVC_RESULT GetWirelessDevicesPowerState(
    int nDevice,
    int * pnState
);
```

Parameters

nDevice

0 : Wifi

1 : Phone

2 : Bluetooth

pnState

0 : Off

1 : On

2 : Discoverable(Bluetooth)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.41 SetWirelessDevicesPowerState

Turning on/off power about WiFi, Phone, Bluetooth

```
SYSSVC_RESULT SetWirelessDevicesPowerState(
    int nDevice,
    int nState
);
```

Parameters

nDevice

0 : Wifi

1 : Phone

2 : Bluetooth

nState

0 : Off

1 : On

2 : Discoverable(Bluetooth)

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.

2.5.42 SetPWR_EN

Enable/Disable PWR_EN pin of RFID port on the backside of the body.

```
SYSSVC_RESULT SetPWR_EN(
    BOOL bEnable,
);
```

Parameters

Enable

True : Enable

False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when other than UHF module is attached to RFID port, so Never use when UHF module is equipped.

2.5.43 SetFW_EN

Enable/Disable FW_EN pin of RFID port on the backside of the body.

```
SYSSVC_RESULT SetFW_EN(  
    BOOL bEnable,  
);
```

Parameters

Enable

True : Enable

False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when other than UHF module is attached to RFID port, so Never use when UHF module is equipped..

2.5.44 SetRF_5V

Enable/Disable RF_5V pin of RFID port on the backside of the body

```
SYSSVC_RESULT SetRF_5V(  
    BOOL bEnable,  
);
```

Parameters

Enable

True : Enable

False : Disable

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully

Notes

It is a function used when a device aside from UHF module is attached to RFID port, so Never use when UHF module is equipped.

In order to communicate with the serial port when a device aside from UHF module is equipped, be sure to enable RF_5V

2.5.45 GetHardwareOptions.

Get the hardware option equipped in PDA

```
SYSSVC_RESULT GetHardwareOptions(  
    HW_OPTIONS *pOptions  
);
```

Parameters

pOptions

the pointer of Structure where hardware option will be stored.

Return Values

SYSSVC_RESULT_SUCCESS will be returned if performed successfully.